REMARKS

This Amendment is in response to the Office Action dated December 2, 2005 ("OA"). In the Office Action, claim 10 was rejected under 35 USC §102 and claims 1-9 were rejected under 35 USC §103. By this Amendment, claims 1, 4-6 and 10 are amended, and new claims 11-16 are added. Currently pending claims 1-16 are believed allowable, with claims 1, 6 and 10 being independent claims.

DOUBLE PATENTING:

The Office Action alleges that claims 6 and 10 are substantially duplicative. OA, pg. 2. The Applicants respectfully submit that both claims recite different subject matter. For example, claim 10 recites, in part, "a third computer readable code embodied in tangible media for assigning an entry in the temporary buffer to the logic source address of said current instruction if the no-dependency signal is not active." This claim element is not present in claim 6. Thus, claims 6 and 10 are not believed to be substantially duplicative.

CLAIM REJECTIONS UNDER 35 USC §101:

Claims 6, 9 and 10 were rejected under 35 USC §101 as allegedly directed to non-statutory subject matter. OA, pg. 2.

Claim 6 is amended to recite, in part, "computer readable code embodied in tangible media." When functional descriptive material is recorded on some computer-readable medium it becomes structurally and functionally interrelated to the medium and will be statutory in most cases since use of technology permits the function of the descriptive material to be realized. MPEP 2106. Thus, the Applicants respectfully submit that claim 6 recites statutory subject matter.

Claim 9 is dependent on and further limits claim 6. Since claim 6 is believed to recite statutory subject matter, claim 9 is also believed to recite statutory subject matter for at least the same reasons as claim 6.

Claim 10 is amended to recite, in part, "computer readable code embodied in tangible media." When functional descriptive material is recorded on some computer-readable medium it becomes structurally and functionally interrelated to the medium and will be statutory in most cases since use of technology permits the function of the descriptive material to

be realized. MPEP 2106. Thus, the Applicants respectfully submit that claim 10 recites statutory subject matter.

CLAIM REJECTIONS UNDER 35 USC §102 AND §103:

Claims 1-9 were rejected under 35 USC §103 as being obvious over U.S. Patent No. 5,778,248 to Leung ("Leung") in view of U.S. Patent No. 5,974,526 to Garg et al. ("Garg"). OA, pg. 4.

Claim 10 was rejected under 35 USC §102 as being anticipated by U.S. Patent No. 5,996,063 to Gaertner et al. ("Gaertner"). OA, pg. 3.

Claim 1

Claim 1 is amended herein to recite, in part, "addressing a mapping-table-entry of a mapping table with a logical source register address of said current instruction thus determining the mapped physical target register address." This claim element was previously found in claims 4 and 5.

As shown in Fig. 9 of the pending Application, the logical source register address 910 is used to access a mapping-table-entry of a mapping table 940. Thus, each entry in the mapping table stores the physical register tag 960 on which the logical register is currently mapped. App., ¶ 69.

The Office Action alleges that Garg contains teaching that describes addressing a mapping-table-entry of a mapping table with a logical source register address of said current instruction thus determining the mapped physical target register address. OA, pg. 6. Specifically, the Office Action contends that such a teaching can be found somewhere in the following sections of Garg:

a. Column 1, line 66 to column 2, line 15

b. Column 3, lines 11-27

c. Column 4, lines 17-40

d. Column 6, lines 48-63

e. Column 12, lines 30-58

f. Figure 1

g. Figure 8

The Applicants respectfully disagree with the Examiner's interpretation of Garg and submit that Garg to does not teach or suggest the recited subject matter either alone or in combination with Leung.

Column 1, line 66 to column 2, line 15 of Garg state,

> When instructions are issued out of order and complete out of order, correspondence between register and values breaks down, and values conflict for register. The problem is severe when the goal of register allocation is to keep as many values in as few registers as possible. Keeping a large number of values in a small number of registers creates a large number of conflicts when the execution order is changed from the order assumed by the register allocator.
>
> Anti- and output dependencies are more properly called "storage conflicts" because reusing storage locations (including registers) causes instructions to interfere with one another even though conflicting instructions are otherwise independent. Storage conflicts constrain instruction issue and reduce performance. But storage conflicts, like other resource conflicts, can be reduced or eliminated by duplicating the troublesome resource. Garg, col. 1, ln. 66 - col. 2, ln. 15.

It is clear from inspection of this passage that no mention or suggestion is made of addressing a mapping-table-entry of a mapping table with a logical source register address of said current instruction thus determining the mapped physical target register address.

Column 3, lines 11-27 of Garg state,

> One technique for removing storage conflicts is by providing additional register that are used to reestablish the correspondence between registers and value. The additional registers are conventional allocated dynamically by hardware and the registers are associated with values needed by the program using "register renaming." To implement register renaming, processors typically allocate a new register for every new value produced (i.e., for every instruction that writes a register). An instruction identifying the original register, for the purpose of reading its value, obtains instead the value in the newly allocated register. Thus, hardware renames the original register identifier in the instruction to identify the new register and correct value. The same register identifier in several different instructions may access different hardware registers, depending on the locations of register references with respect to register assignments. Garg, col. 3, ln. 11-27.

This passage deals with the broad concept of register naming and also does not mention or suggest addressing a mapping-table-entry of a mapping table with a logical source register address of said current instruction thus determining the mapped physical target register address.

Column 4, lines 17-40 of Garg state,

> A further technique for reducing dependencies is using register renaming with a reorder buffer which uses associative lookup. The associative lookup maps the register identifier to the reorder buffer

> entry as soon as the entry is allocated, and, to avoid output dependencies the lookup is prioritized so that only the value for the most recent assignment is obtained if the register is assigned more than once. A tag is obtained if the result is not yet available. There can be as many instances of a given register as there are reorder buffer entries, so there are no storage conflicts between instructions. The values for the different instances are written from the reorder buffer to the register file in sequential order. When the value for the final instance is written to the register file, the reorder buffer no longer maps the register, the register file contains the only instance of the register, and this is the most recent instance.
>
> However, renaming with a reorder buffer relies on the associative lookup in the reorder buffer to map register identifiers to values. In the reorder buffer, the associative lookup is prioritized so that the reorder buffer always provides the most recent value in the register of interest (or a tag). The reorder buffer also writes values to the register file in order, so that, if the value is not in the reorder buffer, the register file must contain the most recent value. Garg, col. 4, ln. 17-40.

This passage deals with associated lookup in a reorder buffer and is silent to the teaching or suggestion of addressing a mapping-table-entry of a mapping table with a logical source register address of said current instruction thus determining the mapped physical target register address.

> Column 6, lines 48-63 of Garg state,
>
> Out-of-order results for several instructions being executed at the same time are stored in a set of temporary buffers, rather that the file register designated by the instruction. If the DDC determines, for example, that a register that instruction 6's source is written to by instructions 2, 3 and 5, then the TAL will indicate that instruction 6 must wait for instruction 5 by outputting the "tag" of instruction 5 for instruction 6. The tag of instruction 5 shows the temporary buffer location where instruction 5's result is stored. It also contains a one bit signal (called a "done flag") that indicates if instruction S is finished or not. The TAL will output three tags for each instruction, because each instruction can have three source registers. If an instruction is not dependent on any previous instruction, the TAL will output the register file address of the instruction's input, rather an a temporary buffer's address. Garg, col. 6, ln. 48-63.

The Applicants respectfully submit that this passage does not mention or suggestion addressing a mapping-table-entry of a mapping table with a logical source register address of said current instruction thus determining the mapped physical target register address.

> Column 12, lines 30-58 of Garg state,
>
> Because it is possible that an instruction might get one of its inputs from a register that was written to by several other instructions, the present invention must choose which one is the real dependency. For example, if instructions 2 and 5 write to register 4 and instruction 7 reads register 4, then instruction 7 has two possible dependencies. In this case, it is assumed that since instruction 5 came after

> instruction 2 in the program, the programmer intended instruction 7 to use instruction 5's result and not instruction 2's. So, if an instruction can be dependent on several previous instructions, RRC 112 will consider it to be dependent on the highest numbered previous instruction.
>
> Once TAL 122 has determined where the real dependencies are, it must locate the inputs for each instruction. In a preferred embodiment of the present invention, the inputs can come from the actual register file or an array temporary buffers 116. RRC 112 assumes that if an instruction has no dependencies, its inputs are all in the register file. In this case, RRC 112 passes the IXS1, IXS2 and IXS/D addresses that came from IFIFO 102 to the register file. If an instruction has a dependency, then RRC 112 assumes that the data is in temporary buffers 116. Since RRC 112 knows which previous instruction each instruction depends on, and since each instruction always writes to the same place in temporary buffers 116, RRC 112 can determine where in temporary buffers 116 an instruction's inputs are stored. It sends these addresses to register file read ports 119 and register file 117 outputs the data from temporary buffers 116 so that the instruction can use it. Garg, col. 12, ln. 30-58.

Similarly, the Applicants submit that this passage does not mention or suggestion addressing a mapping-table-entry of a mapping table with a logical source register address of said current instruction thus determining the mapped physical target register address.

The Applicants further submit that Figures 1 and 8 of Garg do not teach or suggest addressing a mapping-table-entry of a mapping table with a logical source register address of said current instruction thus determining the mapped physical target register address. Furthermore, Leung does not teach or suggest a renaming process.

For at least the reasons given above, claim 1 is believed allowable over the cited references.

Claims 2-5, 11 and 12

Claims 2-5, 11 and 12 are dependent on and further limit claim 1. Since claim 1 is believed allowable, claims 2-5, 11 and 12 are also believed allowable for at least the same reasons as claim 1.

Claim 6

Claim 6 is amended herein to recite, in part, "a fourth computer readable code embodied in tangible media for addressing a mapping-table-entry of a mapping table with a logical source register address of said current instruction thus determining the mapped physical target register address." Support for this claim element can be found in Fig. 9, paragraph 69 and original claims 4 and 5 of the Application.

As discussed in detail above, Garg does not teach or suggest addressing a mapping-table-entry of a mapping table with a logical source register address of said current instruction thus determining the mapped physical target register address. Furthermore, Leung does not teach or suggest a renaming process. Thus, claim 6 is believe allowable over the cited documents.

Claims 7-9, 13 and 14

Claims 7-9, 13 and 14 are dependent on and further limit claim 6. Since claim 6 is believed allowable, claims 7-9, 13 and 14 are also believed allowable for at least the same reasons as claim 6.

Claim 10

Claim 10 is amended herein to recite, in part, "a fifth computer readable code embodied in tangible media for addressing a mapping-table-entry of a mapping table with a logical source register address of said current instruction thus determining the mapped physical target register address." Support for this claim element can be found in Fig. 9, paragraph 69 and claims 4 and 5 of the Application.

As discussed in detail above, Garg does not teach or suggest addressing a mapping-table-entry of a mapping table with a logical source register address of said current instruction thus determining the mapped physical target register address. Furthermore, Leung does not teach or suggest a renaming process. Thus, claim 10 is believe allowable over the cited documents.

Claims 15 and 16

Claims 15 and 16 are dependent on and further limit claim 10. Since claim 10 is believed allowable, claims 15 and 16 are also believed allowable for at least the same reasons as claim 10.

NEW CLAIMS:

New claims 11-16 are added to the present application by this Amendment. No new matter is believed to be introduced by these claims. Specifically, support for the subject matter of claims 11, 13 and 15 can be found at least at paragraph 47 of the Application. Support for the subject matter of claims 12, 14 and 16 can be found at least at paragraph 68 of the Application.

CONCLUSION

In view of the forgoing remarks, it is respectfully submitted that this case is now in condition for allowance and such action is respectfully requested. If any points remain at issue that the Examiner feels could best be resolved by a telephone interview, the Examiner is urged to contact the attorney below.

No fee is believed due with this Amendment, however, should a fee be required please charge Deposit Account 50-0510. Should any extensions of time be required, please consider this a petition thereof and charge Deposit Account 50-0510 the required fee.

Respectfully submitted,

Dated: March 2, 2006

Ido Tuchman, Reg. No. 45,924
Law Office of Ido Tuchman
69-60 108th Street, Suite 503
Forest Hills, NY 11375
Telephone (718) 544-1110
Facsimile (718) 544-8588